p. 28; by the way, this is the definition of a schema type—we are given only its syntax, and the term "binding" is never properly defined). Next, are types attributes of expressions or values or both? The definition suggests that the first assumption is true, but we can also find statements, like: "If x and y are two objects of types t and u respectively, then the ordered pair (x, y) is an object with the *Cartesian product type* $t \times u$" (p. 27). Without a proper definition it is very difficult to answer many important questions, e.g. what does it exactly mean that two expressions "have the same type", or what constructors can be applied to what types.

Chapter 3 is written in a different sytle. It is sytematically organized and describes the "minimal" Z language in full detail. However, as the author says, "For practical use, it needs to be augmented with the basic mathematical definitions in Chapter 4", and "A full grammar for Z is given in Chapter 6" (p. 44). Chapter 4, on the other hand, already uses the Z notation. Hence Chapters 3 and 4 are interdependent. They are hard to read. One has to spend quite some time scanning back and forth to get all needed information. For those who do not know enough mathematics, Chapter 4 will be quite difficult. Part of the problem with these two chapters is in the nature of the subject and cannot be avoided, part in the form of presentation, but part also in the previously mentioned lack of precision in the basic definitions.

I have no special comments on the remaining pages, with one exception: the definitions in the glossary have the same deficiences as those in Chapter 2.

There are some printing errors in the book, but a careful reader should not have a problem in correcting them.

Though I make critical remarks, I am sure this is a useful book despite all its failings. I know people who, after having studied this manual, became Z users and even experts. However, it is a book neither for beginners, nor for those who want just to find out what Z is about. Yet a serious Z user should definitely study this manual. I hope the author prepares a new, improved edition, which would satisfy those with deeper mathematical background and expectations.

Jan Madey
Institute of Informatics
University of Warsaw
Warsaw, Poland

Improving Floating-Point Programming. Edited by Peter J.L. Wallis. Wiley, Chichester, United Kingdom, 1990. Price £29.95 (hardback), ISBN 0-471-92437-7.

The underlying logical properties of floating point have had surprisingly little study, in spite of it being the basis of all numerical calculations on computers for the last 30 years.

This book considers in some depth both classical floating point and also another form of floating point called "Karlsruhe arithmetic". The idea behind this arithmetic is to reduce the propagation of rounding errors when using floating point in a manner which can be implemented effectively in VLSI.

A typical classical floating-point unit might have 24 binary places with an exponent range of −127 to 127 (say). With Karlsruhe arithmetic based upon such a classical floating-point system, one has a very long accumulator so that all the exponent range values can be accommodated. In this case, the mantissa length would be 256 binary places. With such an accumulator, addition into the accumulator loses no precision, and hence a scalar product can be computed with only one rounding error (which occurs when storing the final result).

Each chapter of the book has a separate author, but the book is a coherent whole due to the fact that it is the result of the ESPRIT project DIAMOND. The European Commission is to be congratulated in providing such a permanant record of this project.

The first four chapters are by Wallis and summarise the properties of classical floating point as described in the IEEE standard, the Brown Model and the Ada language standard. The vital chapter is the sixth one which gives the details of Karlsruhe arithmetic.

The problem of embedding the idea within high-level languages is fully covered by means of a dialect of Pascal and special Ada packages. Various methods of optimization are fully considered using Ada as an example.

There is, of course, no doubt that Karlsruhe arithmetic can reduce the effects of rounding error, and that this is more effective than using double length. Also, a hardware implementation is possible, and has been produced on an experimental basis. However, the introduction of such a method must have clear advantages over classical floating point. I have to say that the book contains no examples which could convince me that Karlsruhe arithmetic is anything more than a curious and interesting version of floating point.

The book is well written, contains some useful material, but at nearly thirty pounds, can only be recommended for libraries (when numerical analysis and floating point is of interest).

Brian Wichmann
National Physics Lab
Teddington
Middlesex, UK

The Mathematics of Petri Nets. By Christophe Reutenauer. Prentice Hall International, Hemel Hempstead, United Kingdom, 1990, Price £29.95 (hardback), ISBN 0-13-561887-8.

One of the unwritten laws on mathematics states that the importance, beauty and appeal of a problem is directly proportional to the simplicity of its statement and